AMENDMENTS TO THE CLAIMS

1. (Original) A method of processing work units from client systems comprising:

allocating a plurality of processing slots based on respective priorities of the respective client systems;

assigning work units to the plurality of processing slots; and

sending the work units to the client systems for processing.

2. (Original) The method of claim 1, further comprising:

receiving a high priority work unit from a first client;

pushing current work units of the first client onto a client stack; and

processing the high priority work unit by assigning the high priority work unit to the plurality of processing slots.

3. (Currently Amended) The method of claim 2, wherein allocating comprises[[:]] allocating a predetermined number of processing slots to process high priority work units.

4. (Original) The method of claim 3, wherein the predetermined number of processing slots approximately corresponds to a predetermined portion of the plurality of processing slots, and the plurality of processing slots varies based at least in part on availability of resources.

5. (Original) The method of claim 1, wherein the work units comprise network data packets.

6. (Currently Amended) The method of claim 1, further comprising[[:]] re-prioritizing a first work unit in a client stack based at least in part on a second work unit being at least partially dependent upon the first work unit.

Claims 7-10. (Canceled)

11. (Currently Amended) A method ~~of processing multiple threads~~ comprising:

allocating a plurality of processing slots on a processor based on a priority of ~~the~~ multiple threads;

assigning the multiple threads to the plurality of processing slots;

sending the multiple threads out for processing; and

filling the plurality of processing slots with new threads.

12. (Original) The method of claim 11 further comprising:

receiving a first thread having a high priority;

pushing an older thread on a stack; and

allocating the first thread to the plurality of processing slots.

13. (Currently Amended) The method of claim 12 further comprising[[:]] popping the older thread from the stack when processing resources become available.

14. (Currently Amended) ~~An article comprising: a storage medium which stores instructions, the instructions, when executed, causing systems~~ A machine-readable medium having stored thereon data representing sets of instructions which, when executed by a machine, cause the machine to:

allocate a plurality of processing slots based on a priority of client systems;

assign work units to the plurality of processing slots; and

send the work units to the client systems for processing.

15. (Currently Amended) The ~~article~~ machine-readable medium of claim 14, wherein sets of instructions, when executed by the machine, further cause the machine ~~the storage medium further comprises instructions, the instructions, when executed, cause systems~~ to:

receive a high priority work unit from a first client;

push current work units of the first client onto a client stack; and

process the high priority work unit by assigning the high priority work unit to the plurality of processing slots.

16. (Currently Amended) The ~~article~~ machine-readable medium of claim 14, wherein sets of instructions, when executed by the machine, further cause the machine ~~the storage medium further comprises instructions, the instructions, when executed, cause systems~~ to[[:]] re-prioritize a first work unit in a client stack at least partially based on a second work unit being at least partially dependent upon the first work unit.